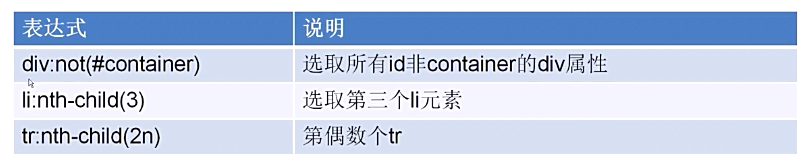
**Scrapy-css选择器**

2017年11月01日 08:16:44

阅读数：1874

和xpath选择器比起来,感觉CSS选择器容易一些,跟写.css时方法基本一样,就是在获取内容时和xpath不同,这里需要注意一下.   



这里介绍如何用css选择器提取出一篇文章的数据   
提取的数据跟xpath那篇文章内容相同   
之前xpath中我们获取元素是通过.entry-header h1::text,如果是属性则用.entry-header a::attr(href)   
介绍一个常用的函数extract\_first()   
相当于extract()[0],但是extract()[0]当数组没有元素时,也就是没有获取到数据时会出错,所以用extract\_first()方法,也可以加上需要返回的内容,比如空,extract\_first("")

title = response.css(".entry-header h1::text").extract\_first()

#p可以不加

create\_date = response.css("p.entry-meta-hide-on-mobile::text").extract()[0].strip().replace('·','').strip()

#获取点赞数

praise\_nums = response.css('#110287votetotal::text').extract()[0]

#获取收藏数

fav\_nums = response.css('.btn-bluet-bigger.href-style.bookmark-btn .register-user-only::text ').extract()[0].strip()

match\_re = re.match('.\*?(\d+).\*',fav\_nums)

if match\_re:

#获取收藏数

fav\_nums = match\_re.group(1)

comment\_nums = response.css('.btn-bluet-bigger.href-style.hide-on-480::text').extract()[0].strip()

match\_re = re.match('.\*?(\d+).\*',fav\_nums)

if match\_re:

comment\_nums = match\_re.group(1)

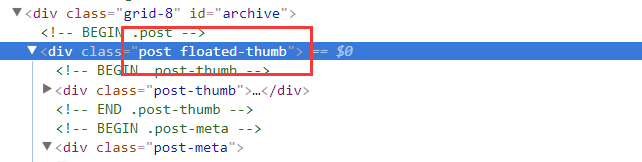
tag\_list = response.css('.entry-meta-hide-on-mobile a::text').extract()

content = response.css('div.entry').extract()[0]

tag\_list = [element for element in tag\_list if not element.strip().endswith('评论')]

tag = ','.join(tag\_list)

当我们要选择的属性名字有多个时比如下面:

   
这市在选择时应该用

post\_urls = response.css('#archive .post.floated-thumb .post-thumb a::attr(href)').extract()

* 1

也就是.post.floated-thumb应该连起来,或者只写.floated-thumb

**完整代码(准)**

def parse\_detail(self, response):

title = response.css(".entry-header h1::text").extract\_first()

create\_date = response.css("p.entry-meta-hide-on-mobile::text").extract()[0].strip().replace("·","").strip()

praise\_nums = response.css(".vote-post-up h10::text").extract()[0]

fav\_nums = response.css(".bookmark-btn::text").extract()[0]

match\_re = re.match(".\*?(\d+).\*", fav\_nums)

if match\_re:

fav\_nums = int(match\_re.group(1))

else:

fav\_nums = 0

comment\_nums = response.css("a[href='#article-comment'] span::text").extract()[0]

match\_re = re.match(".\*?(\d+).\*", comment\_nums)

if match\_re:

comment\_nums = int(match\_re.group(1))

else:

comment\_nums = 0

content = response.css("div.entry").extract()[0]

tag\_list = response.css("p.entry-meta-hide-on-mobile a::text").extract()

tag\_list = [element for element in tag\_list if not element.strip().endswith("评论")]

tags = ",".join(tag\_list)

pass